

<Convex Hull Trick, CHT>

위 Optimization 기법은, 특정 점화식을 만족하는 DP 문제에서 사용 가능

$$\Rightarrow ① dp[i] = \min(dp[j] + b[j] \times a[i]), j \leq i$$

$$② b[j] \geq b[j+1] \text{ (배열 } b \text{ 는 단조감소한다)}$$

$O(n^2)$ 을 $O(n \log n)$ 으로 줄일 수 있고, $a[i] \leq a[i+1]$ 을 만족할 시, $O(n)$ 까지 가능

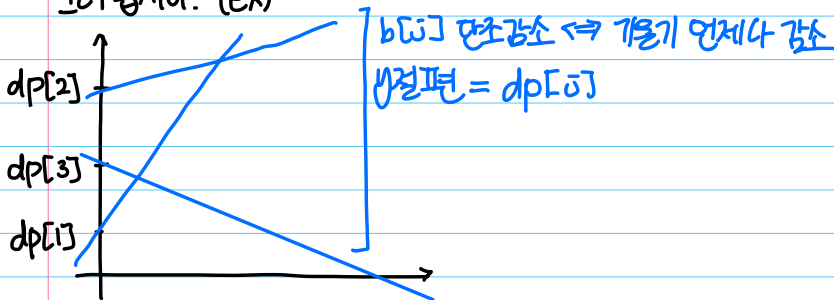
- 점화식에서, $a[i]$ 를 x 로 바꾸면?

$$dp[i] = \min(dp[j] + b[j]x)$$

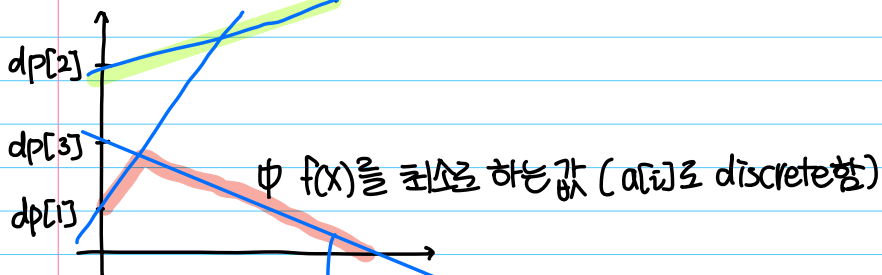
일차함수 아닌가요?

$\rightarrow j$ 가 증가할수록 기울기 \downarrow

- 그려봅시다. (ex)



- 그렇다면, $\min(dp[j] + b[j]x)$ 가 여기서 나타내는 모양은?

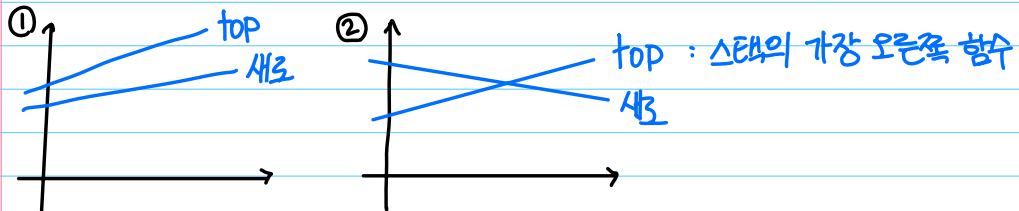


\rightarrow 이게 결국 볼록 껍질 (Convex Hull)이 아니겠는가?
 \rightarrow 아예 구하는 문제로 바꿔준다

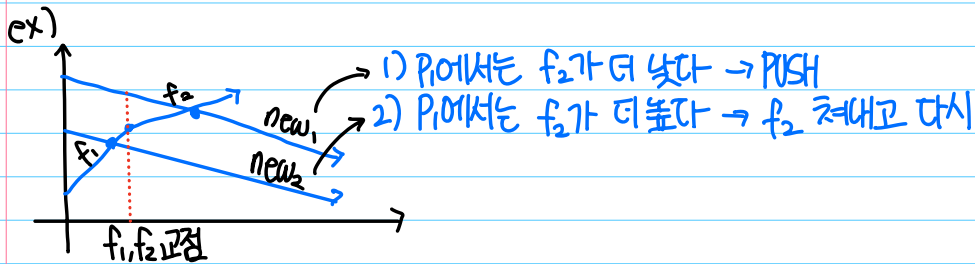
- Convex Hull 방법론

1. $DP[0]$ 계산
2. 스택에 $f(x) = b[0] \cdot x + dp[0]$ 함수 추가
3. $f[AC[i]]$ 최솟값 찾기 ($DP[i]$ 계산)
4. 스택에 $f(x) = b[i] \cdot x + dp[i]$ 함수 추가
- ⋮

- 여기서, 위 그래프의 ●처럼 최솟값과 관련 없는 함수가 들어올 수 있다
- 함수가 들어오는 경우

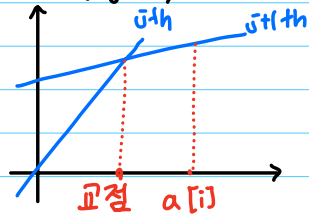


교점으로 구분 가능하다.



- ① top을 삭제할 필요가 있을 때 반복
 - ②, 또는 ① 이후 → 새로 stack에 push
- Convex Hull 알고리즘과 유사

- Q. 최솟값 찾기? Reminder) $dp[i] = \min(dp[j] + b[j] \times a[i])$



$a[i]$ 가 더 크다? \Rightarrow 찾는 함수가 j 번째 이후에 있다.

$\rightarrow f_j(a[i]) < f_{j+1}(a[i])$ 의 논리

위와 같은 방법으로 이상적인 함수를 parametric search 해준다.

<BOJ 13263 나무자르기>

i 번째 나무의 높이는 $a[i]$

오른쪽이 자른 나무 중 최대 길이 $b[i] =$ 전기톱 충전 비용
는 단조증가

j 번째를 자르고 i 번째를 자르는 경우

$\rightarrow dp[i] = \min(0 \leq j < i) (a[i] b[j] + dp[j])$ 로 표현 가능하다!!